Courseware Adaptation for Programming Logic Classes for Teaching Children with Blindness or Severe Low Sight

Angelo Amaral, Camila Aparecida Paris Baston,
Carolina de Carvalho Amaral and
Marcos Augusto Francisco Borges

October 18, 2024

Courseware Adaptation for Programming Logic Classes for Teaching Children with Blindness or Severe Low Sight

Ângelo Amaral [0000-0001-6400-6258] 1, Camila Aparecida Paris Baston [0000-0003-1491-7963] 1, Carolina de Carvalho Amaral [0009-0008-1562-8227] 2, Marcos Augusto Francisco Borges [0000-0003-3580-5178] 1

[1] Universidade Estadual de Campinas (UNICAMP) – Limeira, SP – Brasil
[2] Universidade Estadual Paulista "Julio de Mesquita Filho" (UNESP) – Rio Claro, SP – Brasil
angelo.amaral@gmail.com

Abstract. This article explores the necessity of modifying traditional educational resources used in teaching programming logic to better meet the needs of students with visual impairments. By integrating assistive technologies, blind and severely visually impaired learners are empowered to develop programming skills, thereby promoting a more inclusive education environment. The primary contributions of our work include: (i) a systematic review on programming education for blind children, (ii) the discussion of a toolkit composed of existing assistive software components to support software development by blind students, based on the literature review, and (iii) the adaptation of educational materials for teaching programming logic tailored to the needs of visually impaired students.

Keywords: Programming, Children, Blindness, Visual Impairment, Severe Low Sight.

1 Introduction

Worldwide, 2.2 billion people experience some level of visual impairment, according to the Pan American Health Organization (PAHO/WHO), including 400 million people with severe visual impairment or blindness [1]. At the same time, the International Monetary Fund (IMF) estimates a global shortfall of up to 85 million IT professionals by 2030 [2]. Bridging these realities by facilitating access to programming classes for students with visual impairments presents a valuable opportunity.

Teaching programming languages to students with severe visual impairment or blindness requires enabling access to knowledge and skills previously restricted due to the limitations of traditional education models. By adapting software development interfaces with assistive tools aimed at empowering visually impaired developers, teachers can foster greater inclusion in their classrooms, promoting more diverse and representative groups of future programmers.

Introducing children with severe low sight or blindness to software development through adapted tools and platforms makes teachers mediators of an even greater inclusion process [3], supporting the autonomy of visually impaired students in coding on equal terms with their peers.

This work expands a previous short paper [4] presented in poster format at the 26th International Conference on Human-Computer Interaction (HCI International 2024) and accepted for future publication by Springer Nature. It proposes adapting activities from the didactic materials of a programming logic course, presenting a toolkit based on existing assistive software for guiding students through the early stages of coding [5]. The paper is structured as follows: Section 2 compiles a set of resources for teaching programming to visually impaired children based on a systematic literature review; Section 3 proposes adapting the didactic materials for teaching programming logic; Section 4 suggests a free assistive toolkit to support blind or severely visually impaired programmers, and Section 5 presents the conclusion.

2 Literature Review and Related Work

We analyzed 29 works in a systematic literature review on programming education for visually impaired children. The review methodology followed Kitchenham's guidelines [6], conducted through searches in four different digital libraries for articles published between 2015 and 2024.

The research questions guiding our review were: "What is the state of the art in programming education for students with severe low sight or blindness?" and "Do these resources support children's learning?" To answer these questions, the search string "programming AND (blind OR 'visually impaired')" was used in the ACM Digital Library, IEEE Xplore, SBC Open Library, and the CAPES Journal Database, with the string translated into Portuguese for the latter two.

Of the 52 works returned, the ACM Digital Library provided 23 results, IEEE Xplore returned 15 articles, SBC Open Library provided 4 works, and CAPES Journals resulted in 10 more. We excluded 23 papers — 9 from the ACM Digital Library, 5 from IEEE Xplore, 1 from SBC Open Library, and 8 from CAPES Journals — for not being related to programming education or learning resources for blind or severely visually impaired students.

Among the 29 remaining works, 11 were related to facilitating the programming learning process for visually impaired students through assistive tools, while 18 articles addressed the learning process more broadly and its challenges for blind or visually impaired students. Only 8 of the articles supported children's learning, with 3 works providing existing software toolkits for assisting them in coding [7], [8], [9].

The exploratory analysis classified the works into five objectives: i) Supporting children's learning (Cl), ii) Supporting learning at other ages (Ol), iii) Supporting programming language teaching (Pl), iv) Supporting blind students (Bs), and v) Supporting students with severe low sight (Ls). Furthermore, we identified four application domains during the literature review: i) Use of Software resources (Sr), ii) Use of Hardware resources (Hr), iii) Use of existing assistive resources (Er), and iv) Proposing new assistive resources (Pr).

Based on this literature review, we identified that using existing software tools to assist children with severe visual impairments in learning programming is consistent with the current state of the art. In parallel, most works focused on older students, indicating promising opportunities for future research.

2.1 Analyzed Works

Armaly et al. [10] presented a study focusing on the use of existing assistive software resources to teach blind students programming without considering students with low sight or children. Armaly et al. [11] expanded on this work by addressing new assistive technologies while maintaining the same proposal.

Baker et al. [12] proposed the implementation of new assistive software tools to support blind students in learning programming languages, without targeting children. Baker et al. [13] adopted these previously developed tools, incorporating learning support practices and proposing activities to engage students. Similarly, Schanzer et al. [14] presented the development of assistive software to support the learning process for students with low sight, while Schanzer et al. [15] addressed the adaptation of existing development interfaces to adopt the tools developed in their previous work, also considering blind students.

Oliveira [16] presented a programming language developed to support low-sighted students in learning robotics, without focusing on children, and Oliveira et al. [17] expanded the language's features, detailing application scenarios and compiling the results of their work.

Lunuwilage et al. [18] covered existing software tools to support low-sighted students without considering blind students or children. Johnson et al. [19] added blind students to their work, while Seo et al. [20] adopted a similar approach, focusing only on blind students, not on low-sighted developers. Azevedo et al. [21], Ehtesham-Ul-Haque et al. [22], and Potluri et al. [23] presented similar works covering both low-sighted and blind students.

Villalobos [5] proposed developing new software resources to support low-sighted students' learning, focusing on children. Falase et al. [24], Ludi [25], and Cavazos-Carrizales et al. [26] worked on the same approach, adding hardware customization to their assistive technology proposals, while all these works focused on educational resources to support children's learning.

In parallel, Villar et al. [27] also worked on educational resources and hardware customization without any software customization and without focusing on children. Perin et al. [28] and Robe et al. [29] addressed the adaptation of hardware and software in educational resources for blind and low-sighted students, without specific support for children's learning.

Capel et al. [30] and Blas et al. [31] discussed using existing assistive software resources, focusing on blind students and excluding children, while Konecki et al. [32] focused on low-sighted students rather than blind ones, and Gonçalves et al. [33] worked similarly, aiming at both blind and low-sighted students.

Pires et al. [3] described the adoption of existing assistive software resources to support children with low sight in learning programming languages.

2.2 Related Works

As related works, we considered articles focused on existing software tools to assist blind children in learning programming languages. Rong et al. [7] and Sabuncuoglu [9] presented articles on teaching programming languages to blind and low-sighted children by adopting existing hardware and software assistive resources, proposing customizations, and developing new tools to support them in software development while excluding older students. Milne [8] presented a similarly comprehensive study, although focused only on blind students.

This work compiles existing assistive software resources to support blind and low-sighted children in learning programming languages, creating a toolkit of free applications aimed at enhancing the student's software development experience, as well as proposing adaptations to the didactic materials used in programming logic classes.

3 Courseware Adaptation for Severely Low-Sighted or Blind Students

The development of a child born with a visual impairment is hindered due to the difficulty in contacting and exploring the environment [34]. Thus, during the early years, the sensory integration, synthesis, and interpretation of information provided through other perceptual channels must be extensively explored by parents or guardians of visually impaired children [35] by stimulating other senses for their development, such as the haptic (touch) and auditory systems, in addition to the use of language.

Regarding the need for stimulation, Santaella [36] asserts that the tactile system responds to specific stimuli: i) pressure, ii) heat, iii) cold, iv) pain, and v) kinesthesia, resulting in 1) cutaneous touch restricted to the skin, 2) haptic touch at the junction of the skin with joint movement, 3) dynamic touch involving skin, joints, and muscles, 4) thermal touch associated with vasodilation or vasoconstriction, and 5) directional touch leading to object perception in relation to gravity. Therefore, the haptic system involves the entire body, unlike other more localized perceptual systems.

According to Jacob [37], a visually impaired child uses the resources available to them to relate to their surroundings through sensory stimuli, and although visual impairment presents restrictions concerning perception, it does not prevent the child from creating relationships with their environment. Just like children without visual impairments, their learning process must be active, guided by interest in the subject, and supported by ways of understanding each concept presented and relating the information [38]. Through active learning, children acquire the ability to solve problems, relate information, and transfer acquired knowledge to new situations [39].

In the school environment, visually impaired children require not only accessibility in all spaces but also adaptations of their didactic materials as needed. This may involve the use of magnifying glasses, glasses, and telescopes, representing an improvement in terms of quality, comfort, and visual performance [40], noting that students with severe low sight or blindness also use tactile, textured, and audio materials.

Additionally, the tactile adaptations must consider easy perception through touch and, whenever possible, consist of different textures to better distinguish the components of the whole [40]. Smooth/rough and fine/thick contrasts are adopted because they allow appropriate distinctions, and the material should not cause rejection during handling or be easily damaged due to tactile exploration and constant use. The adapted material must be simple, providing practical use and not posing a danger to students.

According to Lourenço et al. [41], adapting didactic materials for visually impaired students should be based on criteria related to their usability efficiency, including: i) Fidelity to the original model; ii) The material must be pleasant to the touch and/or for auditory reception, with no background noise; iii) Visual stimulation should be based on the appropriate selection of resources, including colors, sounds, and contrast; and iv) Visual limitations resulting from the lack of vision should not be ignored or neglected when developing or selecting materials and teaching resources.

As Gil [42] states, "the hands become the eyes of visually impaired individuals," reflecting the need for intense manual stimulation, encouraged, enhanced, and necessary for the student to develop two skills: i) "educate" the hand so that it represents both a perception organ and an exploration tool, and ii) "develop" bimanual coordination (of both hands) so that the ear/hand coordination substitutes the eye/hand coordination established by individuals without visual impairment.

Based on these premises, the didactic material for the initial programming logic lessons from "Once Upon a Time" project [43] was adapted. The lesson, titled "Programming on Paper," aims to stimulate students' ability to construct algorithms by describing the movements needed to navigate a maze represented by a 4x4 matrix, where only the squares of a certain color can be used in the path, as shown in Figure 1.

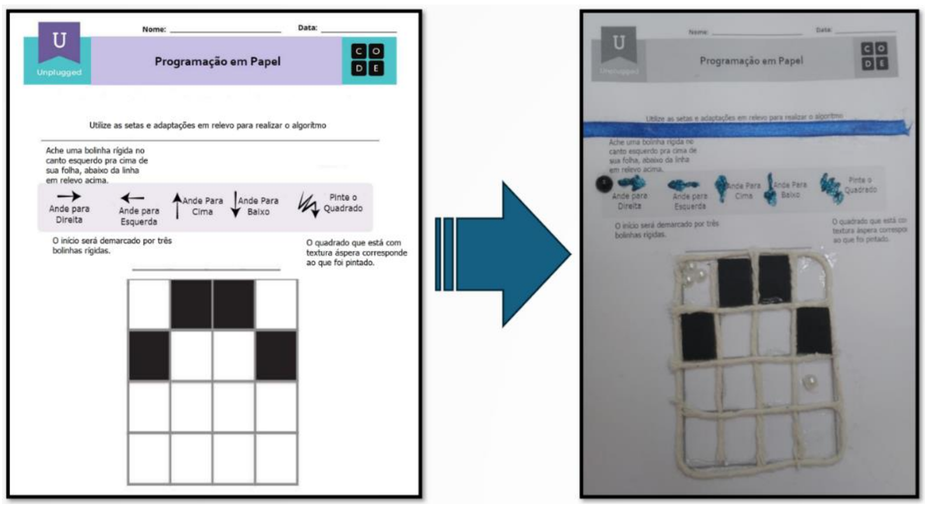

Fig. 1. Initial Adaptation Proposal and First Implemented Version

The first challenge in adapting the material was that visually impaired students cannot rely on instructions based on the color of the squares. It was necessary to replace the concept of "colored squares" with "textured squares" by using sandpaper pieces, as suggested by Gil [42] and Sá et al. [40].

Next, it was essential to address the spatial issue, considering that the student would not be able to identify the matrix represented on paper. This was resolved using string to create the tactile outline of each square and small raised spheres to indicate the maze's entry and exit points.

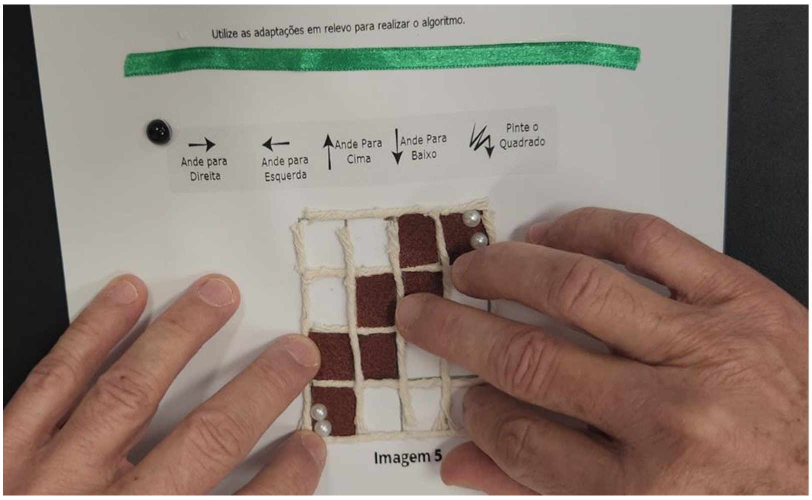

Fig. 2. Second Implemented Version

In the second version, further modifications were made, considering the need for students to identify the starting point in the maze and their path through squares with the same texture. A finer-grain sandpaper was used to provide a softer touch for navigation, as defined by Lourenço [41]. Figure 2 illustrates the version used for model validation.

The adapted activity was validated by a group of 5 teachers at different educational levels, all experienced in teaching technology. The group consisted of i) 1 elementary school teacher experienced in computational thinking and Maker culture activities, ii) 3 high school teachers with experience teaching various technical subjects, and iii) a university professor experienced in teaching technology to undergraduate and graduate students. All teachers were part of the course "Topics in Information Technology: New Approaches and Technologies for Teaching" at UNICAMP's School of Technology.

For a more realistic experience of visual impairment, the teachers were blindfolded during the application of the activity, as illustrated in Figure 3, and had no prior contact with the adapted material.

The peer validation, carried out through the activity with volunteer teachers, allowed for identifying opportunities for teaching programming logic to visually impaired individuals — from blindness to severe low sight — using simple craft materials such as string, sandpaper, and plastic beads to build the didactic materials.

Fig. 3. Model Validation

4 Proposed Toolkit and Application Scenario

According to the World Health Organization [44], blindness varies from low to no residual vision, while low sight is characterized by significant loss of visual acuity in both eyes, as well as field of vision loss. For this study, visual impairment is defined as severe low sight to blindness, i.e., greater visual losses.

Considering that blind programmers cannot rely on assistive tools typically used by low-sighted programmers, such as screen magnifiers [3], [17], [32], this work focuses on complementing the adapted didactic materials with assistive tools to support blind or severely low-sighted children in learning programming skills [4]. These resources can explore other senses, such as screen readers, dictation software, and block-based programming [25], assuming literacy as a prerequisite for defining the target student age range of 10 to 14 years.

Screen magnification tools are outside the scope of this work as they do not support blind or severely low-sighted students, as they only magnify specific screen areas [13]. For this specific impairment, adopting a screen reader that reads the entire screen content to students is the best adaptation [23]. The recommended screen reader is NVDA software [45], a free, open-source solution for Windows that can run from a USB drive without any installation required.

To accelerate code writing [15], the toolkit also includes dictation software to convert voice inputs into text, provided as part of Microsoft Windows' assistive features through Windows 11 Voice Typing [46], a speech-to-text software supporting voice commands for all keyboard inputs and shortcuts in English, Spanish, and German, with limited support for other languages.

To complete the toolkit, we recommend adopting block-based programming languages along with the previous resources, supporting blind children's initial programming learning. These students benefit from predefined blocks composed of basic code blocks [8], [25], which guide them in their first coding steps. We propose adopting Scratch [47], a free programming tool available in over 70 languages.

5 Conclusion

As access to technology is a relevant and modern issue, our work helps teachers support visually impaired children — from blindness to severe low sight, aged 10 to 14 — in their first contact with programming in a diverse and representative environment.

Future steps in this work aim to expand the set of adapted activities to cover the entire material presented in the "Once Upon a Time" project [43], offering a complete introductory programming logic course for blind or severely low-sighted children. We also plan to apply the proposed toolkit with visually impaired elementary school students during the course and evaluate the results to propose best practices. Additionally, we consider extending the solution to include customized lesson plans, expanding the programming course's scope for blind and severely low-sighted students.

Our main contributions are: (i) a systematic review of programming education for blind and severely low-sighted children, (ii) a discussion of a toolkit composed of existing assistive software components designed to support blind or severely low-sighted students' software development, based on the literature review, and (iii) the adaptation of didactic materials for teaching programming logic based on the needs of visually impaired students.

References

[1] PAHO, “Visual Health - PAHO/WHO,” Pan American Health Organization. Visual Health. Accessed: Mar. 26, 2024. [Online]. Available: https://www.paho.org/en/topics/visual-health

[2] P. N. da Costa, “The Global Competition for Technology Talent,” IMF Finance & Development Magazine. International Monetary Fund. Accessed: Mar. 26, 2024. [Online]. Available: https://www.imf.org/en/Publications/fandd/issues/2019/03/global-competition-for-technology-workers-costa

[3] A. C. Pires, F. Rocha, A. J. de Barros Neto, H. Simão, H. Nicolau, and T. Guerreiro, “Exploring accessible programming with educators and visually impaired children,” in *Proceedings of the Interaction Design*

and Children Conference, in IDC ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 148–160. doi: 10.1145/3392063.3394437.

[4] Â. Amaral, C. A. P. Baston, and M. A. F. Borges, “Assistive Toolkit for Teaching Programming to Visually Impaired Children [Unpublished manuscript],” in *HCI International 2024 – Late Breaking Posters*, Cham: Springer Nature Switzerland.

[5] L. P. Q. Villalobos, “Design and Implementation of a Prototype Based in 2D Blocks with Multiple Shapes for Teaching Computer Programming to Visually Impaired Children,” in *2021 IEEE Sciences and Humanities International Research Conference (SHIRCON)*, Nov. 2021, pp. 1–4. doi: 10.1109/SHIRCON53068.2021.9652298.

[6] B. Kitchenham, “Procedures for Performing Systematic Reviews,” *Keele University Technical Report TR/SE-0401*, pp. 1–26, 2004.

[7] Z. Rong, N. fung Chan, T. Chen, and K. Zhu, “CodeRhythm: A Tangible Programming Toolkit for Visually Impaired Students,” in *Proceedings of the Eighth International Workshop of Chinese CHI*, in Chinese CHI ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 57–60. doi: 10.1145/3403676.3403683.

[8] L. R. Milne, “Blocks4All: making block programming languages accessible for blind children,” *SIGACCESS Access. Comput.*, no. 117, pp. 26–29, Feb. 2017, doi: 10.1145/3051519.3051525.

[9] A. Sabuncuoglu, “Tangible Music Programming Blocks for Visually Impaired Children,” in *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*, in TEI ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 423–429. doi: 10.1145/3374920.3374939.

[10] A. Armaly and C. McMillan, “An empirical study of blindness and program comprehension,” in *Proceedings of the 38th International Conference on Software Engineering Companion*, in ICSE ’16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 683–685. doi: 10.1145/2889160.2891041.

[11] A. Armaly, P. Rodeghero, and C. McMillan, “[Journal First] A Comparison of Program Comprehension Strategies by Blind and Sighted Programmers,” in *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, May 2018, p. 788. doi: 10.1145/3180155.3182544.

[12] C. M. Baker, L. R. Milne, and R. E. Ladner, “StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, in CHI ’15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 3043–3052. doi: 10.1145/2702123.2702589.

[13] C. M. Baker, C. L. Bennett, and R. E. Ladner, "Educational Experiences of Blind Programmers," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, in SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 759–765. doi: 10.1145/3287324.3287410.

[14] E. Schanzer, S. Bahram, and S. Krishnamurthi, "Accessible AST-Based Programming for Visually-Impaired Programmers," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, in SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 773–779. doi: 10.1145/3287324.3287499.

[15] E. Schanzer, S. Bahram, and S. Krishnamurthi, "Adapting Student IDEs for Blind Programmers," in *Proceedings of the 20th Koli Calling International Conference on Computing Education Research*, in Koli Calling '20. New York, NY, USA: Association for Computing Machinery, 2020. doi: 10.1145/3428029.3428051.

[16] J. D. Oliveira, "GoDonnie: definição e avaliação de uma linguagem de programação para comandar robô por programadores iniciantes com deficiência visual," Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, 2017.

[17] J. D. Oliveira, M. de B. Campos, A. Amory, and R. H. Bordini, "GoDonnie: A Robot Programming Language to Improve Orientation and Mobility Skills in People Who are Visually Impaired," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, in ASSETS '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 679–681. doi: 10.1145/3308561.3354599.

[18] K. Lunuwilage, S. Abeysekara, L. Witharama, S. Mendis, and S. Thelijjagoda, "Web based programming tool with speech recognition for visually impaired users," in *2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Dec. 2017, pp. 1–6. doi: 10.1109/SKIMA.2017.8294132.

[19] J. Johnson, A. Begel, R. Ladner, and D. Ford, "Program-L: Online Help Seeking Behaviors by Blind and Low Vision Programmers," in *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Sep. 2022, pp. 1–6. doi: 10.1109/VL/HCC53370.2022.9833106.

[20] J. Seo and M. Rogge, "Coding Non-Visually in Visual Studio Code: Collaboration Towards Accessible Development Environment for Blind Programmers," in *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*, in ASSETS '23. New York, NY, USA: Association for Computing Machinery, 2023. doi: 10.1145/3597638.3614550.

[21] P. de Azevedo, V. Vieira, A. Souza, and F. C. Souza, "Uma abordagem para auxiliar estudantes com deficiência visual na modelagem de sistemas: um estudo piloto," in *Anais da IV Escola Regional de Engenharia*

de Software, Porto Alegre, RS, Brasil: SBC, 2020, pp. 57–66. doi: 10.5753/eres.2020.13716.

[22] M. Ehtesham-Ul-Haque, S. M. Monsur, and S. M. Billah, “Grid-Coding: An Accessible, Efficient, and Structured Coding Paradigm for Blind and Low-Vision Programmers,” in *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, in UIST ’22. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3526113.3545620.

[23] V. Potluri, P. Vaithilingam, S. Iyengar, Y. Vidya, M. Swaminathan, and G. Srinivasa, “CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, in CHI ’18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–11. doi: 10.1145/3173574.3174192.

[24] O. Falase, A. F. Siu, and S. Follmer, “Tactile Code Skimmer: A Tool to Help Blind Programmers Feel the Structure of Code,” in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, in ASSETS ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 536–538. doi: 10.1145/3308561.3354616.

[25] S. Ludi, “Position paper: Towards making block-based programming accessible for blind users,” in *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*, Oct. 2015, pp. 67–69. doi: 10.1109/BLOCKS.2015.7369005.

[26] J. P. Cavazos-Carrizales and F. J. Ruiz-Sanchez, “Computer Vision Interface for Symbolic Programming of Cartesian Motion to introduce Visually Impaired Children into Robotic Sciences,” in *2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, Nov. 2022, pp. 1–6. doi: 10.1109/CCE56709.2022.9975943.

[27] N. Villar, C. Morrison, D. Cletheroe, T. Regan, A. Thieme, and G. Saul, “Physical Programming for Blind and Low Vision Children at Scale,” in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, in CHI EA ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–4. doi: 10.1145/3290607.3313241.

[28] A. Perin, D. E. Silva, and N. Valentim, “Um Estudo de Viabilidade sobre a Assistente Virtual ADA Blocks com Professores do Ensino Médio,” in *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*, Porto Alegre, RS, Brasil: SBC, 2022, pp. 1243–1254. doi: 10.5753/sbie.2022.224682.

[29] R. Robe, B. P. Salton, and S. Bertagnolli, “Recursos Pedagógicos para o Ensino de Programação de Estudantes com Deficiência Visual: uma revisão sistemática da literatura,” *Revista Novas Tecnologias na Educação*, vol. 18, no. 1, 2020.

[30] G. Capel, D. Torres, and J. I. Vidal, “The use of learning objects in the academic adequacy in an object-oriented programming course for a blind student,” in *2017 Twelfth Latin American Conference on Learning Technologies (LACLO)*, Oct. 2017, pp. 1–4. doi: 10.1109/LACLO.2017.8120938.

[31] M. J. Blas, M. F. Golobisky, M. Castellaro, and D. A. García Lozano, “Accessibility at the University: An Experience on How Teaching Programming to Blind Students,” in *2018 XIII Latin American Conference on Learning Technologies (LACLO)*, Oct. 2018, pp. 165–172. doi: 10.1109/LACLO.2018.00042.

[32] M. Konecki, N. Ivković, and M. Kaniški, “Making programming education more accessible for visually impaired,” in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2016, pp. 887–890. doi: 10.1109/MIPRO.2016.7522265.

[33] R. S. Gonçalves, R. S. Santana, F. A. Neto, S. C. Benevides, and N. S. dos Santos, “Análise dos Desafios para Programar sem Enxergar: estudo de caso na disciplina Linguagem de Programação 1,” in *Anais Estendidos do XVI Simpósio Brasileiro de Sistemas de Informação*, 2020, pp. 17–20.

[34] A. C. B. da Cunha and S. R. F. Enumo, “Desenvolvimento da criança com deficiência visual (DV) e interacção mãe-criança: algumas considerações,” *Psicologia, saúde e doenças*, vol. 4, no. 1, pp. 33–46, 2003.

[35] T. de A. Souza, V. E. de Souza, M. C. B. Lopes, and S. P. S. Kitadai, “Descrição do desenvolvimento neuropsicomotor e visual de crianças com deficiência visual,” *Arq Bras Oftalmol*, vol. 73, pp. 526–530, 2010.

[36] L. Santaella, *Matrizes da linguagem e pensamento: sonora, visual, verbal: aplicações na hipermídia*. Editora Iluminuras Ltda, 2001.

[37] E. M. Jacob, “Para além do campo da visão: materiais complementares para a educação de crianças cegas e de baixa visão,” *Pensares em Revista*, no. 11, 2017.

[38] RBAC, “Sobre a Aprendizagem Criativa,” Portal da RBAC. Rede Brasileira de Aprendizagem Criativa. Accessed: Mar. 25, 2024. [Online]. Available: https://aprendizagemcriativa.org/sobre-aprendizagem-criativa

[39] M. Lodi and S. Martini, “Computational Thinking, Between Papert and Wing,” *Sci Educ (Dordr)*, vol. 30, no. 4, pp. 883–908, 2021, doi: 10.1007/s11191-021-00202-5.

[40] E. D. de Sá, I. M. de Campos, and M. B. C. Silva, “Atendimento Educacional Especializado–Deficiência Visual,” *Brasília: Ministério da Educação*, 2007.

[41] E. A. G. de Lourenço, *Acessibilidade Para Estudantes com Deficiência Visual: Orientações Para o Ensino Superior*, vol. 1. 2020.

[42] M. Gil, *Deficiência Visual*. Brasília: MEC. Secretaria de Educação a Distância, 2000.
[43] LIAG, "Era Uma Vez," Computação Criativa. LIAG: Laboratório de Informática, Aprendizagem e Gestão. UNICAMP. Accessed: Mar. 21, 2024. [Online]. Available: https://liag.ft.unicamp.br/computacaocriativa/era-uma-vez/
[44] WHO, "ICD-11 for Mortality and Morbidity Statistics," World Health Organization. ICD-11: International Classification of Diseases 11th Revision. Code 9D90: Vision impairment including blindness. Accessed: Mar. 19, 2024. [Online]. Available: https://icd.who.int/browse/2024-01/mms/en#1103667651
[45] NV Access Limited, "About NVDA," NV Access. Accessed: May 22, 2024. [Online]. Available: https://www.nvaccess.org/about-nvda/
[46] Microsoft, "Get the Most out of Voice Typing," Windows Learning Center. Accessed: May 22, 2024. [Online]. Available: https://www.microsoft.com/en-us/windows/learning-center/how-to-use-voice-typing
[47] Scratch Foundation, "Scratch - About," Massachusetts Institute of Technology: Scratch . Accessed: May 22, 2024. [Online]. Available: https://scratch.mit.edu/about

Appendix I – Analyzed and Related Work

Objectives: Supporting children's learning (Cl); Supporting learning at other ages (Ol); Supporting programming language teaching (Pl); Supporting blind students (Bs); Supporting students with severe low sight (Ls). Application Domains: Use of Software resources (Sr); Use of Hardware resources (Hr); Use of existing assistive resources (Er); Proposing new assistive resources (Pr).

Table 1. Analyzed Works

Authors	**Objectives**					**Application Domains**			
	Cl	**Ol**	**Pl**	**Bs**	**Ls**	**Sr**	**Hr**	**Er**	**Pr**
Armaly et al., 2018 [11]		X		X		X		X	
Armaly er al., 2016 [10]		X		X		X		X	
Baker et al., 2019 [13]		X	X	X		X		X	
Baker et al., 2015 [12]		X		X		X			X
Pires et al., 2020 [3]	X		X		X	X		X	
Azevedo et al., 2020 [21]		X		X	X	X			X
Schanzer et al., 2019 [14]		X	X		X	X			X
Schanzer et al., 2020 [15]		X		X	X	X		X	
Capel et al., 2017 [30]		X	X	X		X		X	
Johnson et al., 2022 [19]		X		X	X	X		X	
Seo et al., 2023 [20]		X		X		X		X	
Cavazos-Carrizales et al., 2022 [26]	X		X	X	X	X	X		X
Oliveira et al., 2019 [17]		X	X		X	X		X	X
Oliveira, 2017 [16]		X	X		X	X		X	X

Lunuwilage et al., 2017 [18]		X			X	X		X	
Villalobos, 2021 [5]	X		X		X	X			X
Blas et al., 2018 [31]		X	X	X		X		X	
Konecki et al., 2016 [32]		X	X		X	X		X	
Ehtesham-Ul-Haque et al., [22]		X		X	X	X			X
Villar et al., 2019 [27]	X		X	X	X		X	X	X
Falase et al., 2019 [24]		X		X	X	X	X		X
Perin et al., 2022 [28]		X	X	X	X	X	X	X	
Robe et al., 2020 [29]		X	X	X	X	X	X	X	
Gonçalves et al., 2020 [33]		X	X	X	X	X		X	
Ludi, 2015 [25]	X	X	X	X	X	X	X		X
Potluri et al., 2018 [23]		X		X	X	X		X	X

Table 2. Related Works

Authors	**Objectives**					**Application Domains**			
	Cl	**Ol**	**Pl**	**Bs**	**Ls**	**Sr**	**Hr**	**Er**	**Pr**
Rong et al., 2020 [7]	X		X	X	X	X	X	X	X
Milne, 2017 [8]	X		X	X		X	X	X	X
Sabuncuoglu, 2020 [9]	X		X	X	X	X	X	X	X
This work	X		X	X	X	X		X	